Student Name: **Stud id:** **sect #:** **serial#:**

University of Bahrain College of Information Technology
Department of Computer Science

ITCS332: Concepts of Programming Languages **Quiz#4: Chapter 5_Names**

****

QUESTION ONE: Fill in blanks **[4 pts]**

1) In C++, a new scope is created for every ***function*** or ***block***.
2) In C++, the "`&&`" symbol is bound to "and" operation at ***language design*** time, and a call to a library function "`sqrt`" is bound to the function code in the library at ***link*** time.
3) The main advantage of long names is ***improved readability***. The main disadvantage of long names is ***wasting memory space in the symbol table***.
4) The main advantage of static variables is ***No time overhead for allocation and deallocation***. The main disadvantage of static variables is ***that they do not support recursion (memory space overhead)***.

QUESTION TWO: carefully study the following C++ code and answer the questions below: [7 pts]

a) What will be printed after executing the following C++ code?

```
50) int x = 6, y = 4, f = 2*x ;
51) void main ()
52) {  static double t = -7.5;
53)    int x = 7, y = 11, f = 3*x;
54)    {       //block #1
55)       f++;
56)       cout << x << '\t' << y << '\t'
57)            << f << '\t' << t << endl;
58)          {  //block #2
59)             int y = 30;
60)             int x = 20;
61)             t = t + 5;
62)             f = x + y;
63) //           .............
64)          }
65)             x--; y *= 2; t *= 4; f = x - y;
66)             cout << x << '\t' << y << '\t'
67)                  << f << '\t' << t << endl;
68)    }
69) }
```

```
7   11   22   -7.5
6   22  -16   -10
```

b) The scope of the variable x defined in line#53 is ***from line #53 to line #69 except lines #60 to #64***.
c) The referencing environment at line#63 is ***x,y of block#2 and f of main***.
d) The lifetime of the variable t defined in line#52 begins when ***The function main is loaded*** and ends when ***The function main is exited***.

Student Name: **Stud id:** **sect #:** **serial#:**

University of Bahrain College of Information Technology

Department of Computer Science

ITCS332: Concepts of Programming Languages **Quiz#4: Chapter 5_Names**

****

QUESTION ONE: carefully study the following C++ code and answer the questions below: [7 pts]

a) What will be printed after executing the following C++ code?

```
20) int x = 4, y = 6, f = 3*x ;
21) void main ()
22) {  static double t = -4.5;
23)    int x = 11, y = 7, f = 2*x;
24)    {       //block #1
25)       f++;
26)       cout << x << '\t' << y << '\t'
                << f << '\t' << t << endl;
27)        {    //block #2
28)           int y = 20;
29)           int x = 30;
30)           t = t -3;
31)           f = x - y;
32) //         .............
33)        }
34)           x++; y *= 2; t *= 2; f = x + y;
35)           cout << x << '\t' << y << '\t'
                   << f << '\t' << t << endl;
36)    }
37) }
```

```
11   7   23   -4.5
12   14  26   -15
```

b) The scope of the variable x defined in line#23 is **from line #23 to line #37 except lines from #28 to #33**.

c) The referencing environment at line#32 is **x,y of block#2 and t,f of main**.

d) The lifetime of the variable *y* defined in line#28 begins whenever **The block #2 is entered** and ends when **The block #2 is exited.**

QUESTION TWO: Fill in blanks **[4 pts]**

5) In programming languages, a data type specifies: **range of possible values assigned to variables** and **a set of operations applicable on those variables**.

6) Static variables are allocated storage when the **defining program unit/block is loaded for the first time** and deallocated when **the entired program unit/block is exited**.

7) With static type binding languages (such as FORTRAN), the type of a variable is specified using **explicit declaration statement** or **implicit declaration.**

8) The main disadvantage of stack-dynamic variables is **Time overhead for allocation and deallocation**. The main advantage of stack-dynamic variables is **Efficient use of memory space**.

9) The lifetime of a stack-dynamic variable begins whenever **THE DEFINING PROGRAM UNIT/ BLOCK IS ENTERED** and ends whenever **THE DEFINING PROGRAM UNIT/ BLOCK IS EXITED.**

10) In a static-scoped language, the initializing expression of a variable must contain **constants/predefined constant names**. In a dynamic-scoped language, the initializing expression of a variable must contain **variables/constants**.

11) In C++, the lifetime of a variable defined inside a block begins whenever **the block is entered** and ends whenever **the block is exited**.

12) In languages that use dynamic type binding, the type of a variable is specified using a **referencing context** or whenever **it is assigned values during execution**.

13) For every assignment to a subrange variable, the **RANGE CHECKING** is done at run-time, and the **TYPE CHECKING** is done at compile time.

14) In a static-scoped language; referencing environment depends on **the textual layout** of program unit/blocks. In a dynamic-scoped language, the referencing environment depends on **the calling sequence** of program units.

15) Static variables are bound to storage during the **LOADING** time. A data type such as **bool** is bound to a range of possible values during the **LAMGUAGE IMPLEMENTATION** time.

16) The lifetime of an explicit-heap dynamic variable begins whenever **the defining statement/operator is executed** and ends whenever **the deallocating/deleting statement is executed.**

Write a C++ code that illustrates the use of stack-dynamic variables.
Write a C++ code that illustrates the use of explicit heap dynamic variables.